UNCLASSIFIED

JADS JT&E-TR-98-006

JADS JT&E

High Level Architecture Runtime Infrastructure Test Report

By: Maj D.L.Wright, Mr. Clyde J. Harris, Mr. Jerry W. Black

JADS Joint Test & Evaluation • 11104 Menaul N.E. • Albuquerque, NM 87112-2454

UNCLASSIFIED

Executive Summary

Joint Advanced Distributed Simulation (JADS) is an Office of the Secretary of Defense-sponsored joint test force chartered to determine the utility of advanced distributed simulation (ADS) technology for test and evaluation (T&E) of military systems. JADS is doing this by looking at three slices of the T&E spectrum. One of those slices is the JADS Electronic Warfare (EW) Self-Protection Jammer (SPJ) Test. The EW test was the only JADS test that was in a position to look at the new Department of Defense (DoD) standard technical architecture for DoD simulations -- high level architecture. The JADS EW SPJ Test uses high level architecture (HLA) federations to replicate all elements of an actual open air range (OAR) test environment and the selected EW system under test (an ALQ-131 Block II SPJ). To determine the utility of ADS technology for EW T&E, JADS will use and evaluate the HLA as part of the SPJ three-phase test program.

In developing and implementing an HLA federation for EW T&E, JADS recognized that measuring and controlling the latency imposed by diverse test facilities, simulators, communications equipment, and long-haul communications networks was a critical factor. Because of the importance to T&E, most of these latency measurements have been made in other EW test projects or communications architectures and are documented. A new element used by JADS for EW T&E is the HLA and runtime infrastructure (RTI) software. Since the RTI provides a new means for dissimilar simulators and facilities to communicate, an additional source of latency is imposed on a test architecture which must be measured, optimized, and controlled for accurate real-time measurement of test events for comparison with the range data. This effort was undertaken for the JADS EW Test and is the subject of this special report.

The primary objective of JADS RTI testing is to ensure that the EW test has an acceptable communications infrastructure, including the RTI, for each ADS test phase in order to accurately recreate the critical interactions from the OAR test environment. Acceptable means that all hardware and software components are behaving as required and that the total system latency is within budget over the expected range of message rates and sizes used to recreate the OAR test event interactions. After several months of testing and tuning the available RTI parameters, the RTI host computer hardware and operating system, and the network infrastructure, JADS was able to produce an acceptable communications infrastructure for the ADS-based test phases. This report outlines the testing JADS used, the problems JADS encountered, and the lessons that JADS learned during this effort. These results, problems, and lessons are an indication of the current state of the HLA, tools that are available to federation developers, and the RTI software. HLA is still maturing. As new versions of the RTI become available many of the specific measures and some of the problems JADS resolved (discussed in this report) will become obsolete. However, the methodology and the basic approach to testing communications infrastructure latency are independent of the RTI and will remain valid for the foreseeable future.

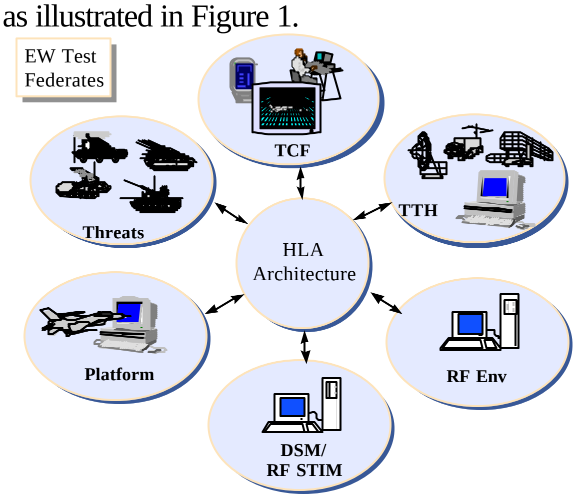
Contents

Figures

1. JADS Electronic Warfare Test Description

Joint Advanced Distributed Simulation (JADS) is an Office of the Secretary of Defense-sponsored joint test force chartered to determine the utility of advanced distributed simulation (ADS) technology for test and evaluation (T&E) of military systems. JADS is doing this by looking at three slices of the T&E spectrum -- one of those slices is the JADS Electronic Warfare (EW) Self-Protection Jammer (SPJ) Test. The JADS EW SPJ Test will use high level architecture (HLA) federates to replicate all elements of an actual open air range (OAR) test environment and the selected EW system under test (an ALQ-131 Block 2 SPJ). The use of the HLA by the Department of Defense (DoD) was directed by the Under Secretary of Defense for Acquisition and Technology (USDA&T) on September 10, 1996, as the standard technical architecture for all DoD simulations. To determine the utility of ADS technology for EW T&E, JADS will use and evaluate the HLA in a three-phase test program.

The OAR test (Phase 1) is a flight test on an instrumented range using an F-16 with a SPJ. The radio frequency (RF) environment, the threat systems, and the jammer are all instrumented to calculate standard EW measures of performance from the data collected. The engagement will be carefully scripted and recreated for use in the Phase 2 and Phase 3 tests, which will use HLA. The purpose of Phase 2 and Phase 3 tests is to gather data to evaluate the utility of ADS using the same test scenario with HLA. JADS will also determine how well the ADS test results correlate with the OAR test results collected in Phase 1. During the ADS test phases, each OAR test run will be recreated using HLA-compliant federations consisting of software models and hardware-in-the-loop (HITL) threat simulators. The federate interactions will be monitored, and the measures of performance will be calculated in real time. A key operating component supporting the JADS test federations is software developed by the Defense Modeling and Simulation Organization (DMSO) called the runtime infrastructure or RTI. Use of the RTI is one of the requirements to be HLA compliant. There are six federates comprising the JADS EW Test federation, as illustrated in Figure 1.



DSM = digital system model
TCF = test control federate
Env = environment
TTH = terminal threat hand-off federate
STIM =radio frequency stimulator

Figure 1. JADS EW Test Federate

In developing and implementing an HLA federation for EW T&E, JADS recognized that measuring and controlling the latency imposed by diverse test facilities, simulators, communications equipment, and long-haul communications networks was a critical factor. Because of the importance to T&E, most of these latency measurements have been made in other EW test projects or communications architectures and are documented. A new element used by JADS for EW T&E is the HLA and, in particular, RTI software. Since the RTI provides a new means for dissimilar simulators and facilities to communicate, an additional source of latency is imposed on a test architecture which must be measured, optimized, and controlled for accurate real-time measurement of test events for comparison with the range data. This effort was undertaken for the JADS EW Test and is the subject of this report. The first step in the process was for JADS EW to define the RTI performance requirements for the Phase 2 and Phase 3 tests.

2. Runtime Infrastructure Test Objective

The primary objective of JADS RTI testing is to ensure that the EW test has an acceptable communications infrastructure, including the RTI, for each ADS test phase (which use the RTI) in order to accurately recreate the critical interactions from the OAR test environment. Acceptable means that all hardware and software components are behaving as required and that the total system latency is within budget over the expected range of message rates and sizes used to recreate the OAR test event interactions.

RTI test results have been provided on a regular basis to DMSO. JADS conducted RTI tests to satisfy two key requirements:

- Quantitatively measure latency and expected RTI 1.3 software performance prior to JADS EW Phase 2 and Phase 3 tests
- Provide input to the verification, validation, and accreditation (VV&A) process for JADS EW Phase 2 and Phase 3 tests

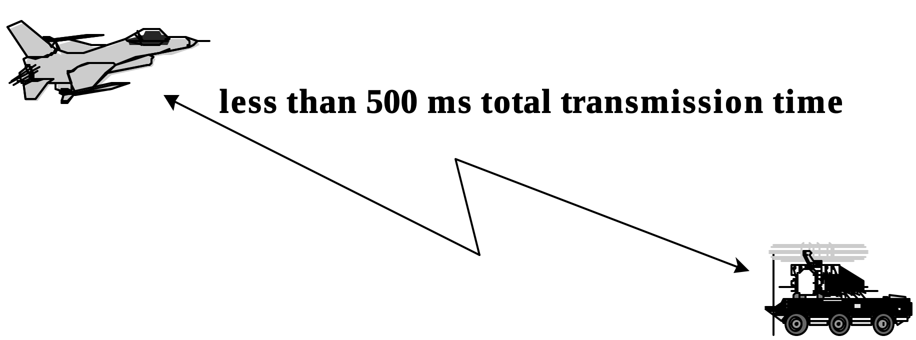
Based on the results obtained, JADS will make minor modifications to the use of RTI services, the data structures, update rates, sizes, or other aspects of the infrastructure necessary to meet the total end-to-end interaction time requirements described in Section 3 below for the Phase 2 and Phase 3 tests.

JADS has participated in the Simulation Interoperability Standards Organization (SISO), has been a member of the Architecture Management Group (AMG) hosted by DMSO for more than two years and has found little applied experience in testing and tuning performance oriented federations in either forum. We believe testing and tuning is necessary for VV&A of the test architecture and should be planned for in the development and implementation of future high-performance federations through a series of tests. Future T&E users of HLA may find useful the test tools and methods described in this report.

3. RTI Performance Requirements for JADS EW Test

The RTI performance requirements definition process we used came from a solid understanding of the interactions between aircraft carrying self-protection jammers and surface-to-air threat systems in an OAR test. The problem space was defined by the reference test condition (RTC) used in the OAR test described in the JADS EW Program Level Test Activity Plan and Data Management and Analysis Plan, dated March 1998. Closed-loop testing using ADS technology runs the risk that the communications infrastructure transmitting the data between federates will change the outcome either through lost interactions or by changing the temporal nature of the exchange. This temporal change is usually an increase in the time for the exchange called latency. The amount of allowable latency depends on the nature of the interactions and the decision cycle of each system involved. The EW test interaction of interest is the threat radar activation, jammer identification and response, and associated threat response.

We focused on determining how much latency the jammer/threat interaction could tolerate and still be valid. Depending on how the engagement is carried out, the interaction can be the jammer's computer working against the threat's computer or the jammer's computer working against the threat's human operator. The latency is driven by the decision cycle times of the jammer computer and either the threat computer or the threat operator. The jammer used in the JADS test is simple and has a very short decision cycle. Likewise the threat computers have very short decision cycles. The analysis showed that it was unrealistic to model the computer-to-computer interaction. The latency expected from linking the Air Force Electronic Warfare Environment Simulator (AFEWES) in Fort Worth, Texas, and the Air Combat Environment Test and Evaluation Facility (ACETEF) at Patuxent River, Maryland, independent of additional elements (e.g., crypto, routers, RTI, etc.) was too great to faithfully reproduce the engagements that normally occur at distances shorter than 50 kilometers (km). In fact, the analysis indicated that once the wide area network (WAN) communications time, the local area network (LAN) communications time, and the facility interface processing times for both AFEWES and ACETEF were accounted for, the acceptable latency for the RTI had to be a negative value. The decision cycle time for the threat operator was estimated to be 500 milliseconds (ms), which we believe is an achievable latency objective for JADS. Therefore, the limitations that we have placed on the communication infrastructure latency with human operator interaction is 500 ms.



Once the total latency was identified, the 500 ms were allocated to the communications infrastructure, facility interfaces, and the RTI. That means from the time the radar changes state, the infrastructure has no more than 500 ms to get that message to the jammer (processing time not included), have it process that message, and then return the jammer's response. We refer to this as an "end-to-end interaction" during the EW test. Of the 250 ms, the RTI is allocated 70 ms, as computed below.

In the ADS environment, the network will add additional latencies to the real latencies described above. Phase 3 of the EW test uses the system under test (SUT) installed in the ACETEF anechoic chamber which is the most complex ADS architecture JADS EW will use. For this configuration, the following steps occur in the ADS environment:

Step 1) Radar on at threat
Step 2) Radar state passed to AFEWES application program interface (API)
Step 3) AFEWES API passes radar state to ACETEF API using RTI reliable transport
Step 4) ACETEF API passes radar state to the Advanced Tactical Electronic Warfare Environment Simulator (ATEWES) to radiate radar RF
Step 5) Jammer initiates a response
Step 6) Jammer instrumentation captures response and transmits to ACETEF API
Step 7) ACETEF API passes jammer state to AFEWES API using reliable transport
Step 8) AFEWES API passes jammer state to the JammEr Techniques Simulator (JETS) to initiate RF
Step 9) Radar receives jammer response

Steps 2, 3, 4, 6, 7 and 8 introduce additional latency to the real-world exchange. Steps 3 and 7 are latencies introduced by the RTI and the geographical latency due to separation of facilities. The expected JADS EW latencies which are the non-RTI latencies are given below:

Step 2 - 50 ms
Step 4 - 100 ms
Step 6 - 60 ms
Step 8 - 50 ms
Total - 260 ms

For reliable data transfer of JADS federation object model (FOM) data types, it is assumed that there will be one transfer to the sending federate's RTI "reliable distributor" software and one transfer from the receiving reliable distributor and RTI to the destination federate for both Steps 3 and 7. This introduces 4 times the expected geographical latency for both RTI latencies (i.e., two geographical latencies per RTI transfer). Based on the HLA Engineering Protofederation data, the geographical latency was measured as 25 ms (one way) between ACETEF and AFEWES. The third JADS facility is located at Albuquerque, New Mexico. The location of the RTI executive and federation executive will be determined by future performance tests once the WANs are installed between the three test nodes.

The total non-RTI latency is therefore 260 ms + 4 * 25 ms = 360 ms.

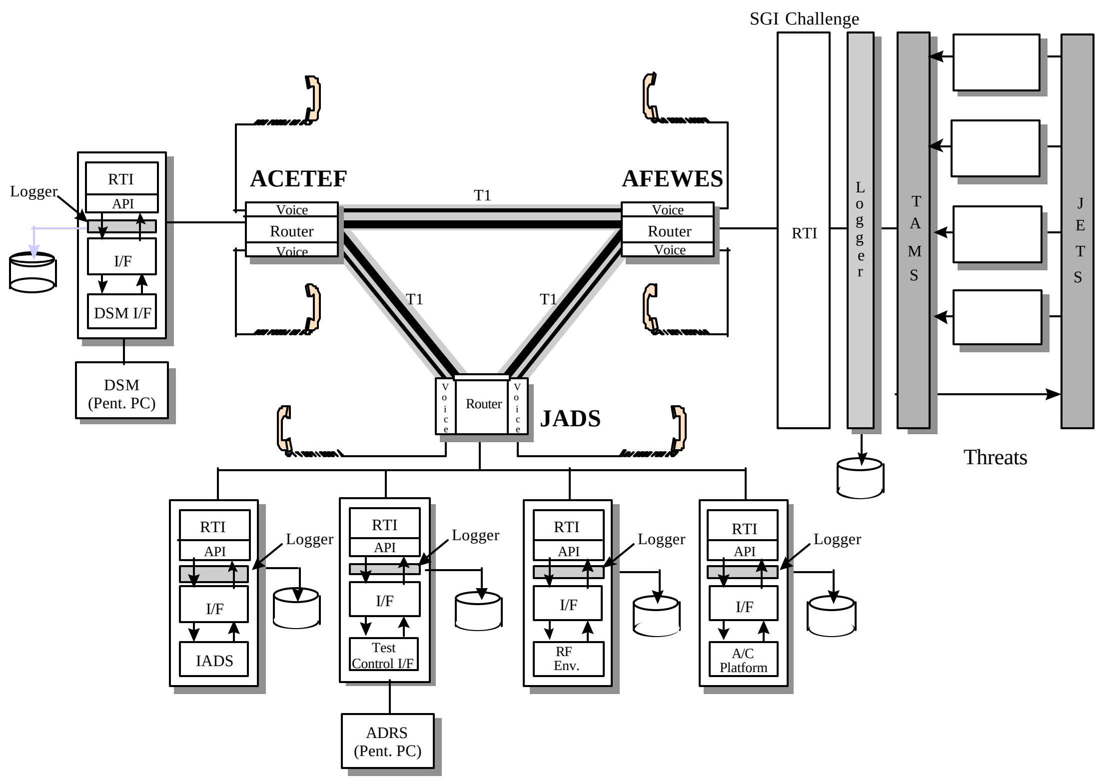
The maximum allowable latency is driven by the time necessary to initiate jamming when a radar is activated, and the time necessary to terminate jamming when a radar beam is pulled off of the target. The most critical time factor for initiating jamming is if the technique is designed to deny acquisition by the threat. As stated previously, the jamming must be presented to the radar within 500 ms. This value is based on the human response time (200 ms for visual recognition + 300 ms for physical reaction) to the technique. In the instance when the radar beam is pulled off the target, the jamming must terminate before the operator can reacquire the jamming signal. This time is again based on human response time of 500 ms as described above. Based on the above requirements, the sum of the two, one-way RTI latencies in Steps 3 and 7 must be less than 500 ms - 360 ms = 140 ms. The maximum one-way RTI latency is therefore 70 ms. The RTI latency is defined as follows:

Step 1) API_{in} to RTI (e.g., AFEWES passes radar state)
Step 2) RTI to RTI over network (e.g., using RTI reliable transport)
Step 3) RTI to API_{out} (e.g., to ACETEF API)

All network latencies between Steps 1-2 and Steps 2-3 have been included in the geographical latencies described above.

4. JADS Federation and Network Description

The JADS EW Test uses dedicated T-1 circuits, communications, and encryption devices to link JADS with two key EW test facilities, AFEWES and ACETEF, in two different states. Three network nodes interconnect a total of six federates representing critical components of the OAR test environment including the test aircraft, aircraft EW systems, and threat systems. Four of the six federates execute on dedicated Silicon Graphics, Inc. (SGI) O2 workstations in the JADS test control facility at Albuquerque, New Mexico. There is one federate executing on an SGI O2 at the ACETEF and one federate executing on an SGI Challenge at the AFEWES HITL facility. The federates at Albuquerque will publish a combined 2 attributes at 20 Hertz (Hz). The worst case instance of the AFEWES federate will have 11 attributes published at 20 Hz. The ACETEF federate will publish 1 attribute at 20 Hz. All nodes will publish interactions at approximately 1 Hz. The largest JADS federation attribute or interaction is 106 bytes in length. One execution of the JADS federation replicating a pass on the OAR will take about four minutes. The JADS test bed used the same computer and communications components that will be installed for the Phase 2 test. The Phase 2 network architecture and test federation are illustrated in Figure 2.



A/C = aircraft
ADRS = Automated Data Reduction Software
DSM = digital system model
Env = environment
I/F = interface
IADS = Integrated Air Defense System
PC = personal computer
TAMS = Tactical Air Mission Simulator

Figure 2. JADS EW Phase 2 Test Architecture and Federates

The following is a summary of the requirements derived for the JADS EW Test federations used for Phase 2 and Phase 3.

Performance Measure	JADS Requirement
Attribute/Interaction Size	Max: 672 bits Min: 16 bits
Update Frequency	Max: 20 Hz Min: 1 Hz
Expected Bandwidth	Max: 183335 bits per second
Time to Create New Objects	10 ms
Central Processing Unit (CPU) Utilization	RTI: 25% Overhead: 5%
Allowable RTI Latency	< 140 ms for closed-loop interaction

Figure 3. Summary of RTI Requirements

The primary tool for documenting and communicating requirements to DMSO and the RTI development community is the Federation Execution Planners Workbook. The JADS EW Federation Execution Planners Workbook is provided as Attachment 2. The workbook contains extensive descriptions of the JADS federates, attributes and interactions, computers and communication equipment and RTI services required. JADS began working with DMSO to articulate our test design and requirements for RTI performance in May 1997 in order to reduce risk to the JADS test in using the RTI and provide the required information to RTI developers.

The hardware used for the RTI tests as well as the JADS EW Phase 2 and Phase 3 tests :

- SGI O2 R5000 (200 megahertz [MHz]) workstation - 2 each
- SGI O2 R10000 (180 MHz) workstation - 4 each
- 5-port 10Base-T hub (generic; for the initial network and RTI tests)
- 8-port 10Base-T/100Base-TX Ethernet switch (CentreCOM FS 708; for recent network and RTI tests)
- KIV-7 crypto - 6 each
- Vera-Link Access System 2000 DLS 2100 channel service unit (CSU)/data service unit (DSU)
- IDNX Micro-20
- 2-port Ethernet router card (Cisco 11.0)
- RS422 serial trunk card
- Voice card
- IDNX-20 - 3 each
- 2-port Ethernet router card (Cisco 11.0)
- RS422 serial trunk card
- Voice card
- Network General packet "sniffer"
- Fireberd 6000A Communications Analyzer

The installation of this hardware is illustrated in Figure 17 in Section 7.

5. Test Software

There are two types of software developed for the JADS RTI tests. First, we developed software to send data one way between two computers. There are versions of this software that perform "raw" network tests (both transmission control protocol [TCP] and internet protocol [IP] multicast) and versions that perform RTI tests. The purpose of the test software is to characterize the network and the RTI in the simplest of cases. The second type of software we developed was an RTI federate capable of running in different configurations on multiple computers within a federation execution. The purpose of this software is to determine how the RTI performs in a more realistic environment under loads anticipated for the JADS Phase 2 and Phase 3 federations.

In all of our tests, latency and lost data are the two metrics we examined. To track lost data, all of our messages (either attributes or interactions) contain a serial number. To calculate latency, the send time is included in the message. When a message arrives, the receive time is saved with the send time to be used to calculate the latency.

It is important to note that this latency measures delays from the time at which each message is time tagged in the sending application software to the time it is received by the final application software, but not delays on the sending side that may occur before then. In other words, the "send time" stored is the time the message was actually passed down to the network software or to the RTI, not the time the message should have been passed down to those layers for a periodic sequence of messages or time critical, one-time-event message. However, for periodic messages, latencies before the time tagging can be detected by creating a histogram of the differences between successive send times. Latency problems appear in this histogram as a movement, broadening, and/or distortion of the distribution of the time differences compared to the expected histogram, which should show a narrow, symmetrical distribution around a nominal difference value determined by the basic message period. Large latency problems show up in the histogram as outliers with time differences well outside the main distribution.

For this design to work, the simulation time for all the computers that participate in a test must be synchronized. For some simulations, this may be the system time of the computers themselves, while in other cases, an external source provides the simulation time to each computer. In the JADS test federation, we will be using as an external source BANCOMM global positioning system (GPS) cards that accept an Inter-Range Instrumentation Group (IRIG) B or GPS input to synchronize the time. Since these cards were not available when we began RTI testing, we used Version 3-5.91 of the Network Time Protocol (NTP) software to synchronize the system clocks on all of our test computers. This public domain software is described in internet "Request for Comment" (RFC) 1305 (Reference 1).

We have a GPS receiver that provides time to one of the SGI O2 computers via its serial port. This computer is the NTP Stratum-1 time server. All of the other computers in the test bed's network receive their time from the time server via the NTP xntpd software. It takes a few days to get the whole system initially configured and settled down. But after that, the system time on all computers remains within 1 ms of GPS time. The xntpd software generates statistics on how well it is keeping time. We used a BANCOMM card to verify that the offset reported by xntpd was accurate and stable.

6. Two-Node Test Description

The RTI test hardware configurations progressively increase in complexity until the entire federation and network architecture (except for T-1 lines) are in place in the JADS test bed. Starting with a simple two computer, point-to-point configuration, we gathered basic performance data for network IP multicast data, network TCP, RTI 1.0-2 best effort, RTI 1.0-2 reliable, RTI 1.3 beta (1.3b) best effort, RTI 1.3b reliable, RTI 1.3-2 Early Access Version (RTI 1.3-2 EAV) reliable, and RTI 1.3-2 (early official release) reliable.

Figure 4 shows the two-node test configuration. The test configuration included all network components using a two-node network for the same series of tests. The associated communications link throughput and latency, and the hardware/software configuration used is also being tested. All sources of possible latency were measured through a disciplined process of adjusting one variable at a time and collecting recorded time data for the same periodic test message transaction in differing reference test conditions. The two-node network test used an SGI O2 5000 and an SGI O2 10000 running the IRIX 6.3 operating system. The test software and RTI were hosted on each computer for all tests using this configuration.

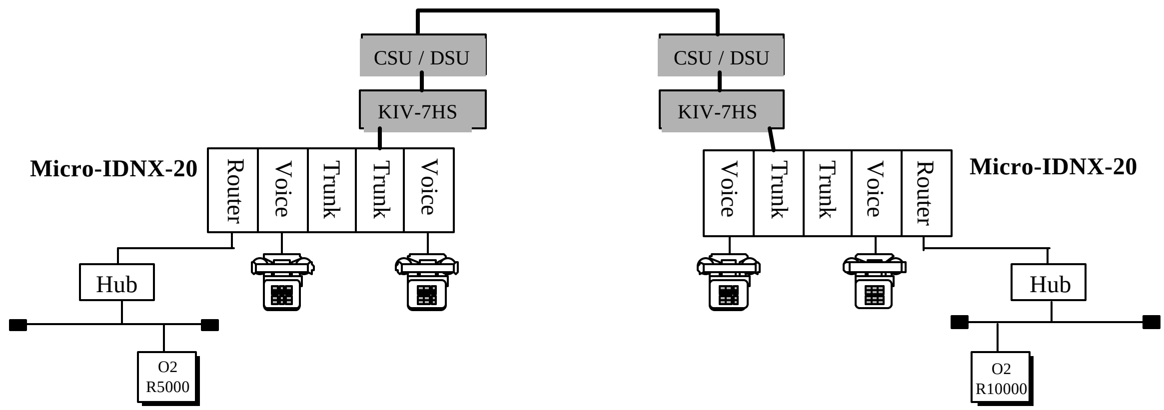


Figure 4. Two-Node RTI Test Configuration with Communications Devices

6.1 Standard Test Methodology for Two-Node Test

Step 1) Baseline hardware configuration performance without RTI
Step 2) Install RTI software
Step 3) Run attribute size tests, attribute rate tests, interaction size, RTI polling interval (and duration) tests using best effort transport with multicasting
Step 4) Add network communications hardware configuration
Step 5) Repeat Steps 1 through 4 for second configuration
Step 6) Compare latency data for different hardware/RTI software configurations

Attribute and interaction message rates, sizes, and tick were each examined around the values specified in the JADS Federation Execution Planners Workbook.

6.2 One-Way Software for Two-Node Tests

The one-way software is designed to exercise the network and the RTI with different data sizes and transmit rates. The size is varied among 17, 51, 101, 301, 501, and 1001 bytes with odd sizes to avoid any standard buffer sizes. The transmit rate is varied among 5, 10, 20, 50, 100, 200, 400, and 500 Hz. The complete matrix of rate and size combinations was tested. Each test case, which consisted of a rate and size pair, ran for thirty seconds. For the RTI version of the one-way software, a separate matrix was generated for attributes sent as reliable and best effort.

There are two programs that must be run in the one-way, network-only (i.e., no RTI) tests – a sender and a receiver. The programs used for these JADS tests are *tcp_sender, tcp_receiver, ipmc_sender,* and *ipmc_receiver*. To generate a test matrix, first start the receiver on one computer. Then, start the sender on another computer. (The *tcp_sender* program requires that the user specify as the destination the host name of the computer upon which the receiver is running.) The sender then loops through each test case of size and rate, sending data to the receiver. At the start of each test case, the sender transmits a **start** message to the receiver indicating the size, rate and total count of messages to be sent. This information is used by the receiver to name the output file and to determine if any messages were lost. After sending the control message, the sender transmits the data messages. Each data message contains a sequential serial number and the time the message was passed down to the underlying network software to be sent. When a message arrives at the receiver, the system time on that computer is obtained. The receiver stores the time sent and time received in an array indexed by the serial number. After sending all of the data for a test case, the sender transmits an **end** message.

When the receiver gets the **end** message, all the data from the test case are written to the data file. To eliminate its effect on the latency calculation, no input/output (I/O) to that file occurs while the data are being transmitted. The data file contains a record for each message that should have been received. If the message was received, the serial number, send time, receive time, and latency are written to the file. Prior to each test case, the receiver initializes the start times to zero. At the end of a test case, if the send time is zero for a serial number, that message was not received. In this case, the serial number and the word MISSING are written to the output file. The receiver also creates a summary file. There is a record in the summary file for each test case run. The record contains the data filename followed by the minimum, maximum, and mean latency for the test case. These simple statistics are often insufficient to accurately describe complex latency events that may occur during a test case, but they can alert the data analyst to trends in the data and to test cases that should be analyzed in more detail.

This sequence of steps is repeated in a test run for every combination of size and rate. Because some of the high data rate and size combinations may disrupt the network, the sender process waits 5 seconds between test cases. When all test cases have been run, an additional **end** message is transmitted by the sender to the receiver to indicate that the test is done.

There is only one federate program used for the one-way RTI tests. It is called *test*. It accepts command line parameters that tell it to run as either the master (-m) federate which initiates data or the slave (-s) federate which only reflects data. To generate an RTI test matrix, first start *test* as a slave on one computer. After a message is displayed that the slave is waiting for data, start *test* as the master on another computer. The processing steps for the *test* federate are the same as the steps for the network tests. It produces data files and a summary file in the same format as the network software.

6.3 One-Way Test Results

Figure 5 shows the network IP multicast test matrix. There were no lost messages until the sender began sending 301byte messages at 500 Hz. These data reflect the performance of the two-node test configuration without the RTI software installed.

Minimum Latency (sec)

Rate	17	51	101	301	501	1001
	Packet Size					
5	0.007	0.007	0.008	0.009	0.011	0.015
10	0.007	0.007	0.008	0.009	0.011	0.015
20	0.007	0.007	0.008	0.009	0.011	0.015
50	0.007	0.007	0.008	0.009	0.011	0.015
100	0.007	0.007	0.008	0.009	0.011	0.015
200	0.007	0.007	0.008	0.009	0.011	0.015
400	0.007	0.007	0.008	0.009	0.011	0.015
500	0.007	0.007	0.008	0.009	0.011	0.015

Maximum Latency (sec)

Rate	17	51	101	301	501	1001
	Packet Size					
5	0.008	0.007	0.008	0.010	0.011	0.015
10	0.007	0.008	0.008	0.010	0.011	0.015
20	0.008	0.008	0.008	0.010	0.012	0.015
50	0.008	0.008	0.008	0.009	0.011	0.015
100	0.009	0.008	0.010	0.013	0.013	0.018
200	0.009	0.008	0.009	0.012	0.012	0.455
400	0.008	0.009	0.011	0.010	0.243	0.456
500	0.010	0.010	0.045	0.172	0.241	0.456

Mean Latency (sec)

Rate	17	51	101	301	501	1001
	Packet Size					
5	0.007	0.007	0.008	0.009	0.011	0.015
10	0.007	0.007	0.008	0.009	0.011	0.015
20	0.007	0.007	0.008	0.009	0.011	0.015
50	0.007	0.007	0.008	0.009	0.011	0.015
100	0.007	0.007	0.008	0.009	0.011	0.015
200	0.007	0.007	0.008	0.009	0.011	0.436
400	0.007	0.007	0.008	0.009	0.235	0.448
500	0.007	0.007	0.008	0.132	0.236	0.448

Values within the border indicate expected rates and sizes for the JADS EW Test
Shading indicates where packets were lost

Figure 5. IP Multicast Test Matrix

Figure 6 shows the network TCP test matrix. The results indicate that there is a significant increase in the latency once the sender transmits at rates greater than 5 Hz. There are also large variations between the minimum and maximum latencies.

Minimum Latency (sec)

	Packet Size					
Rate	**17**	**51**	**101**	**301**	**501**	**1001**
5	0.007	0.007	0.007	0.009	0.010	0.014
10	0.007	0.007	0.008	0.009	0.011	0.014
20	0.007	0.007	0.008	0.009	0.011	0.014
50	0.007	0.007	0.008	0.009	0.011	0.015
100	0.007	0.007	0.008	0.009	0.011	0.015
200	0.007	0.007	0.008	0.009	0.011	0.015
400	0.007	0.007	0.008	0.009	0.011	0.014
500	0.007	0.007	0.008	0.009	0.011	0.015

Maximum Latency (sec)

	Packet Size					
Rate	**17**	**51**	**101**	**301**	**501**	**1001**
5	0.022	0.021	0.022	0.026	0.029	0.035
10	0.206	0.206	0.209	0.211	0.214	0.119
20	0.207	0.208	0.210	0.215	0.169	0.174
50	0.208	0.210	0.215	0.132	0.090	0.088
100	0.209	0.215	0.208	0.178	0.193	0.218
200	0.212	0.159	0.088	0.117	0.128	0.386
400	0.217	0.213	0.054	0.181	0.392	0.393
500	0.214	0.085	0.114	0.085	0.473	0.383

Mean Latency (sec)

	Packet Size					
Rate	**17**	**51**	**101**	**301**	**501**	**1001**
5	0.008	0.008	0.009	0.011	0.013	0.017
10	0.111	0.110	0.111	0.113	0.116	0.090
20	0.104	0.105	0.107	0.114	0.076	0.051
50	0.108	0.110	0.114	0.066	0.050	0.041
100	0.109	0.115	0.078	0.040	0.033	0.033
200	0.112	0.076	0.047	0.033	0.028	0.369
400	0.118	0.047	0.034	0.025	0.364	0.372
500	0.093	0.044	0.032	0.024	0.370	0.372

Values within the border indicate expected rates and sizes for the JADS EW Test

Figure 6. TCP Test Matrix

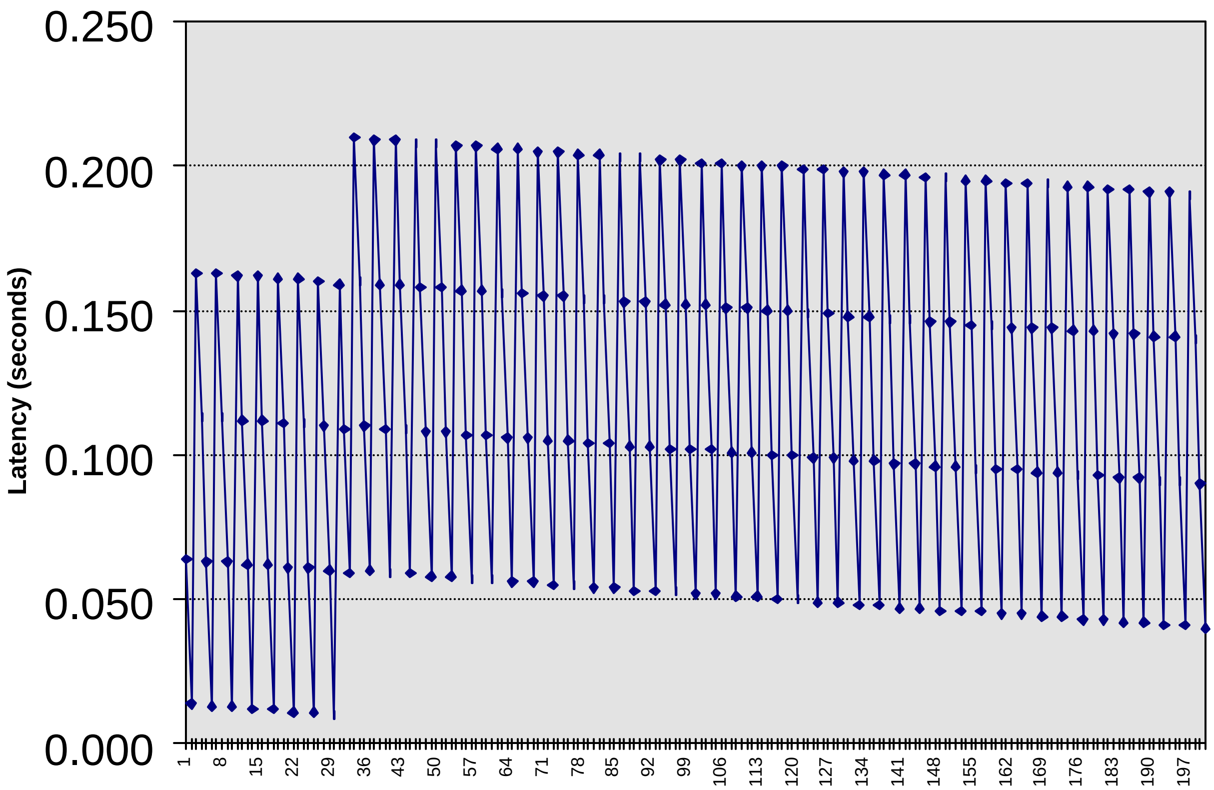


Figure 7. TCP Latency for 101 Bytes at 20 Hz

It is clear from a plot of the data from one trial (see Figure 7) that the data are being buffered somewhere in the transmission path. Upon further investigation, we determined that the buffering was caused by implementation of the Nagle algorithm. The Nagle algorithm, which is described in detail in Reference 2, buffers small packets on the transmit side until an acknowledgment packet (ACK) is received from the previous transmit. On SGI computers, the network can wait up to 200 ms before sending the buffered packets. This explains the jump in latency at transmit rates over 5 Hz. By default, TCP sockets on SGIs run with the Nagle algorithm. To disable the Nagle algorithm, the programmer must specify TRUE for the socket option TCP_NODELAY. Figure 8 shows the network TCP test matrix with the Nagle algorithm disabled.

Minimum Latency (sec)

Rate	17	51	101	301	501	1001
5	0.007	0.007	0.007	0.009	0.010	0.014
10	0.007	0.007	0.007	0.009	0.010	0.014
20	0.007	0.007	0.007	0.009	0.010	0.014
50	0.007	0.007	0.007	0.009	0.010	0.014
100	0.007	0.007	0.007	0.009	0.010	0.014
200	0.007	0.007	0.007	0.009	0.010	0.015
400	0.007	0.007	0.007	0.009	0.011	0.014
500	0.007	0.007	0.007	0.009	0.011	0.015

(Packet Size spans columns 17–1001)

Maximum Latency (sec)

Rate	17	51	101	301	501	1001
5	0.007	0.007	0.008	0.009	0.011	0.015
10	0.007	0.007	0.008	0.010	0.011	0.015
20	0.007	0.007	0.009	0.009	0.012	0.030
50	0.007	0.008	0.009	0.012	0.063	0.181
100	0.008	0.009	0.013	0.011	0.017	0.020
200	0.010	0.013	0.024	0.098	0.146	2.918
400	0.016	0.013	0.018	0.019	3.101	3.235
500	0.011	0.082	0.013	2.914	3.110	3.269

(Packet Size spans columns 17–1001)

Mean Latency (sec)

Rate	17	51	101	301	501	1001
5	0.007	0.007	0.008	0.009	0.011	0.014
10	0.007	0.007	0.007	0.009	0.010	0.014
20	0.007	0.007	0.007	0.009	0.011	0.014
50	0.007	0.007	0.007	0.009	0.010	0.015
100	0.007	0.007	0.007	0.009	0.010	0.014
200	0.007	0.007	0.007	0.009	0.011	0.538
400	0.007	0.007	0.007	0.009	0.571	0.536
500	0.007	0.007	0.007	0.595	0.555	0.533

(Packet Size spans columns 17–1001)

Values within the border indicate expected rates and sizes for the JADS EW Test

Figure 8. TCP Test Matrix with the Nagle Algorithm Disabled

Figure 9 shows the RTI 1.0-2 best effort test matrix. The latencies were slightly higher than the network IP multicast tests. Just as in the multicast tests, the receiver began to lose data when the sender began transmitting 301 bytes at 400 Hz.

Minimum Latency

	Packet Size (bytes)					
Rate (Hz)	17	51	101	301	501	1001
5	0.009	0.009	0.009	0.011	0.012	0.016
10	0.009	0.009	0.009	0.011	0.013	0.016
20	0.009	0.009	0.009	0.011	0.013	0.016
50	0.009	0.009	0.009	0.011	0.013	0.016
100	0.009	0.009	0.009	0.011	0.012	0.016
200	0.009	0.009	0.009	0.011	0.012	0.017
400	0.009	0.009	0.009	0.011	0.013	0.017
500	0.009	0.009	0.009	0.011	0.247	0.309

Maximum Latency

	Size (bytes)					
Rate (Hz)	17	51	101	301	501	1001
5	0.009	0.009	0.010	0.011	0.013	0.017
10	0.010	0.010	0.011	0.012	0.014	0.017
20	0.011	0.010	0.010	0.013	0.019	0.020
50	0.011	0.014	0.013	0.024	0.019	0.018
100	0.017	0.017	0.011	0.038	0.018	0.021
200	0.014	0.015	0.019	0.020	0.018	0.490
400	0.032	0.029	0.021	0.037	0.273	0.488
500	0.788	1.177	1.122	1.123	0.720	0.492

Mean Latency

	Size (bytes)					
Rate (Hz)	17	51	101	301	501	1001
5	0.009	0.009	0.010	0.011	0.013	0.016
10	0.009	0.009	0.010	0.011	0.013	0.016
20	0.009	0.009	0.010	0.011	0.013	0.016
50	0.009	0.009	0.010	0.011	0.013	0.016
100	0.009	0.009	0.010	0.011	0.013	0.016
200	0.009	0.009	0.010	0.011	0.012	0.468
400	0.009	0.009	0.010	0.015	0.263	0.478
500	0.015	0.024	0.023	0.032	0.272	0.481

Shading indicates where packets were lost
Data within the border indicates expected JADS rates and sizes

Figure 9. RTI 1.0-2 Best Effort Test Matrix

Figure 10 shows the RTI 1.0-2 reliable test matrix. Once again, the data shows the effects of the Nagle algorithm in this version of the RTI. However, the latencies are much higher than for the TCP network tests.

Minimum Latency

Rate (Hz)	Size (bytes) 17	51	101	301	501	1001
5	0.009	0.009	0.009	0.011	0.012	0.016
10	0.009	0.009	0.010	0.011	0.013	0.017
20	0.009	0.009	0.010	0.012	0.013	0.017
50	0.009	0.010	0.010	0.012	0.013	0.017
100	0.009	0.010	0.010	0.011	0.013	0.017
200	0.009	0.010	0.010	0.012	0.013	0.017
400	0.009	0.010	0.010	0.012	0.013	0.017
500	0.028	0.079	0.043	0.032	0.032	0.024

Maximum Latency

Rate (Hz)	Size (bytes) 17	51	101	301	501	1001
5	0.023	0.010	0.024	0.012	0.030	0.020
10	0.392	0.378	0.375	0.392	0.392	0.320
20	0.392	0.392	0.392	0.418	0.292	0.315
50	0.392	0.392	0.383	0.239	0.280	0.416
100	0.414	0.273	0.309	0.233	0.170	0.164
200	0.396	0.273	0.400	0.397	0.181	0.359
400	0.396	0.389	0.312	0.233	0.370	0.276
500	0.987	0.996	0.658	1.058	1.096	1.132

Mean Latency

Rate (Hz)	Size (bytes) 17	51	101	301	501	1001
5	0.010	0.009	0.011	0.011	0.015	0.016
10	0.292	0.291	0.289	0.292	0.299	0.204
20	0.291	0.292	0.292	0.263	0.190	0.141
50	0.294	0.294	0.177	0.125	0.161	0.101
100	0.238	0.177	0.191	0.137	0.095	0.070
200	0.246	0.177	0.185	0.138	0.096	0.071
400	0.245	0.177	0.187	0.137	0.096	0.070
500	0.246	0.179	0.187	0.141	0.099	0.074

All packets sent were received
Data within the border indicates expected JADS rates and sizes

Figure 10. RTI 1.0-2 Reliable Test Matrix

Figure 11 shows the RTI 1.3beta (1.3b) best effort test matrix. RTI 1.3b was the first of the RTI version 1.3 software releases we tested. Data loss occurred with smaller packet sizes than the 1.0-2 tests. This was because RTI 1.3b data packet headers were 400 bytes long.

Minimum Latency

Rate (Hz)	17	51	101	301	501	1001
5	0.010	0.011	0.011	0.012	0.014	0.018
10	0.010	0.010	0.011	0.012	0.013	0.017
20	0.010	0.010	0.011	0.012	0.013	0.017
50	0.010	0.010	0.010	0.012	0.013	0.017
100	0.010	0.010	0.010	0.012	0.013	0.017
200	0.010	0.010	0.010	0.012	0.013	0.018
400	0.010	0.010	0.010	0.032	0.066	0.018
500	0.010	0.010	0.010	0.228	0.313	0.469

Maximum Latency

Rate (Hz)	17	51	101	301	501	1001
5	0.013	0.012	0.014	0.015	0.016	0.020
10	0.013	0.013	0.013	0.014	0.017	0.020
20	0.013	0.013	0.062	0.016	0.017	0.020
50	0.016	0.021	0.014	0.016	0.017	0.121
100	0.014	0.015	0.021	0.024	0.017	0.033
200	0.026	0.019	0.065	0.019	0.022	0.543
400	0.101	0.110	0.139	0.252	0.330	0.613
500	1.673	1.700	1.629	1.119	0.810	0.552

Mean Latency

Rate (Hz)	17	51	101	301	501	1001
5	0.011	0.011	0.011	0.013	0.014	0.018
10	0.010	0.011	0.011	0.012	0.014	0.017
20	0.010	0.011	0.011	0.012	0.014	0.017
50	0.010	0.010	0.011	0.012	0.014	0.019
100	0.010	0.011	0.011	0.012	0.014	0.018
200	0.011	0.011	0.011	0.012	0.014	0.522
400	0.011	0.012	0.012	0.233	0.319	0.531
500	0.089	0.079	0.067	0.259	0.333	0.537

Size (bytes)

Shading indicates where packets were lost
Data within the border indicates expected JADS rates and sizes

Figure 11. RTI 1.3b Best Effort Test Matrix

Figure 12 shows the RTI 1.3b reliable test matrix. The effects of the Nagle algorithm are still noticeable here. It wasn't until after we ran the RTI 1.3b tests that we discovered the problem with the Nagle algorithm and how to disable it.

Minimum Latency

Rate (Hz)	17	51	101	301	501	1001
5	0.011	0.011	0.011	0.013	0.014	0.018
10	0.017	0.012	0.018	0.018	0.019	0.033
20	0.020	0.015	0.012	0.018	0.019	0.019
50	0.020	0.016	0.012	0.018	0.022	0.024
100	0.018	0.015	0.013	0.021	0.021	0.017
200	0.014	0.022	0.018	0.016	0.016	0.024
400	0.019	0.037	0.041	0.063	0.090	15.088
500	0.040	0.039	0.037	8.636	23.902	54.706

Size (bytes)

Maximum Latency

Rate (Hz)	17	51	101	301	501	1001
5	0.290	0.214	0.204	0.275	0.215	0.293
10	0.372	0.393	0.365	0.393	0.404	4.760
20	0.393	0.393	0.393	0.335	0.345	0.408
50	0.295	0.391	0.398	0.403	0.269	0.242
100	0.276	0.289	0.275	0.236	0.238	0.137
200	0.252	0.244	0.234	0.146	0.287	16.245
400	0.136	0.164	0.174	9.489	24.223	56.431
500	0.915	0.983	1.315	24.986	36.648	56.511

Size (bytes)

Mean Latency

Rate (Hz)	17	51	101	301	501	1001
5	0.115	0.045	0.090	0.073	0.046	0.106
10	0.271	0.276	0.263	0.286	0.273	0.548
20	0.281	0.283	0.285	0.193	0.186	0.152
50	0.174	0.192	0.192	0.173	0.156	0.103
100	0.143	0.160	0.153	0.113	0.097	0.070
200	0.124	0.114	0.102	0.077	0.063	7.919
400	0.088	0.080	0.074	4.397	12.034	41.853
500	0.099	0.099	0.116	17.417	32.213	55.403

Size (bytes)

All packets sent were received
Data within the border indicates expected JADS rates and sizes

Figure 12. RTI 1.3b Reliable Test Matrix

We provided our RTI 1.3b results to DMSO along with the information we learned regarding the Nagle algorithm and the TCP_NODELAY socket option. DMSO responded to our comments and modified the RTI to disable the Nagle algorithm for all reliable traffic. In addition, they incorporated into RTI

1.3-2early access version (EAV) other modifications intended to improve performance of reliable traffic. Figure 13 shows the RTI 1.3-2EAV reliable test matrix. With the Nagle algorithm disabled, the performance of reliable traffic dramatically improved. However, when the master federate tried to publish 301 byte messages at 400 Hz, reliable data was lost, which is not allowed by the TCP protocol. In addition, when the master federate tried to publish 501 bytes at 400 Hz, the slave federate crashed. These problems never occurred in previous versions of the RTI. However, they are outside the range of the JADS expected performance so we did not concentrate on the specific cause.

Minimum Latency (sec)

Transmit	Packet Size: 17	51	101	301	501	1001
5	0.008	0.008	0.009	0.010	0.012	
10	0.008	0.008	0.009	0.010	0.012	
20	0.008	0.008	0.009	0.010	0.012	
50	0.008	0.008	0.009	0.010	0.012	
100	0.008	0.008	0.009	0.010	0.012	
200	0.008	0.008	0.009	0.010	0.012	
400	0.008	0.008	0.009	0.011		
500	0.008	0.008	0.009	6.516		

Maximum Latency (sec)

Transmit	Packet Size: 17	51	101	301	501	1001
5	0.187	0.010	0.011	0.012	20.922	
10	0.010	0.010	0.011	0.013	0.046	
20	0.010	0.011	0.011	0.012	0.018	
50	0.016	0.018	0.011	0.018	0.138	
100	0.018	0.011	0.024	0.163	0.025	
200	0.011	0.025	0.012	0.120	0.047	
400	0.083	0.020	0.030	15.587		
500	0.139	0.085	0.027	32.477		

Mean Latency (sec)

Transmit	Packet Size: 17	51	101	301	501	1001
5	0.010	0.009	0.009	0.011	7.384	
10	0.008	0.009	0.009	0.011	0.012	
20	0.008	0.009	0.009	0.011	0.012	
50	0.008	0.009	0.009	0.011	0.012	
100	0.008	0.009	0.009	0.011	0.012	
200	0.008	0.009	0.009	0.011	0.012	
400	0.008	0.009	0.009	6.245		
500	0.009	0.009	0.009	12.128		

Values within the border indicate expected rates and sizes for the JADS EW Test
Shaded area indicates data was lost.
Slave crashed during 501 bytes at 400 Hz

Figure 13. RTI 1.3-2EAV Reliable Test Matrix

Figure 14 shows the RTI 1.3-2 reliable test matrix.

Minimum Latency (sec)

Rate	17	51	101	301	501	1001
5	0.008	0.008	0.008	0.010		
10	0.008	0.008	0.008	0.010		
20	0.007	0.008	0.008	0.010		
50	0.007	0.008	0.008	0.010		
100	0.007	0.008	0.008	0.010		
200	0.007	0.008	0.008	0.010		
400	0.007	0.008	0.008			
500	0.007	0.008	0.008			

Maximum Latency (sec)

Rate	17	51	101	301	501	1001
5	0.009	0.009	0.009	0.011		
10	0.031	0.009	0.014	0.011		
20	0.009	0.010	0.013	0.039		
50	0.011	0.010	0.010	0.013		
100	0.011	0.017	0.015	0.023		
200	0.086	0.014	0.019	0.012		
400	0.037	0.019	0.073			
500	0.023	0.057	0.170			

Mean Latency (sec)

Rate	17	51	101	301	501	1001
5	0.008	0.008	0.009	0.010		
10	0.008	0.008	0.009	0.010		
20	0.008	0.008	0.008	0.010		
50	0.008	0.008	0.008	0.010		
100	0.008	0.008	0.008	0.010		
200	0.008	0.008	0.008	0.010		
400	0.008	0.008	0.009			
500	0.008	0.008	0.010			

Values within the border indicate expected rates and sizes for the JADS EW Test
Slave had problems receiving 301 bytes at 400 Hz

Figure 14. RTI 1.3-2 Reliable Test Matrix

This one-way RTI test produced several events with a maximum latency exceeding 70 milliseconds as well as a few smaller events. Our examination of the test data suggests that these latency events can be divided into three classes on the basis of two factors. The first factor is the number of consecutive sample numbers (i.e., test messages) in an event for which the latency exceeds a fixed threshold. It is a rough measure of the seriousness of the latency event. The threshold can be a specific value such as 70 milliseconds derived from the JADS EW Phase 2 and Phase 3 test requirements or a value equal to the mean latency plus a multiple of the latency standard deviation (computed without including the latency events themselves) for each message rate and packet size that would indicate unusual behavior within a test case.

The second factor is the sample number at which the event occurs, i.e., its position with respect to the first sample transmitted by the sender for that message rate and packet size. It divides the events into those that occur soon after the start of message transmission and those that occur later at random times. This factor was suggested by similar event behavior observed in the "raw" TCP/IP latency tests.

The class of isolated events in which the latency exceeds the fixed threshold for only one sample may not be important, since the maximum latency observed during the one-way RTI test for this class was only 39 milliseconds (for a message rate of 20 messages/second and a packet size of 301 bytes). However, we must note that the results shown in Figure 14 represent only one repetition of the one-way test.

Latency events in the other two classes typically follow a pattern of an abrupt transition from the mean latency level to a much higher value that is almost always the maximum latency value for the event, then a gradual decay of the latency values back to the mean level. Figure 15 illustrates this behavior for the largest latency event observed during the one-way RTI test, which occurred for a message rate of 500 Hz and a packet size of 101 bytes (outside of JADS federation requirements). For this event, the latency jumped from the mean level of about 8 milliseconds at sample #15 to the maximum latency value of 170 milliseconds at sample #16. The latency then remained above 70 milliseconds until sample #142, about one-quarter second later. Similar events produced the maximum latency value of 86 milliseconds at the message rate of 200 and a packet size of 17; 73 ms for a message rate of 400 and a packet size of 101; and several smaller values at other rates and sizes. The jagged appearance of the latency plot from the peak until about sample #150 is due to variations in the message receive times. The underlying cause for those variations is not yet known, but it may be due to the details of how the receiving TCP processes data and/or to operating system scheduling of the slave federate.

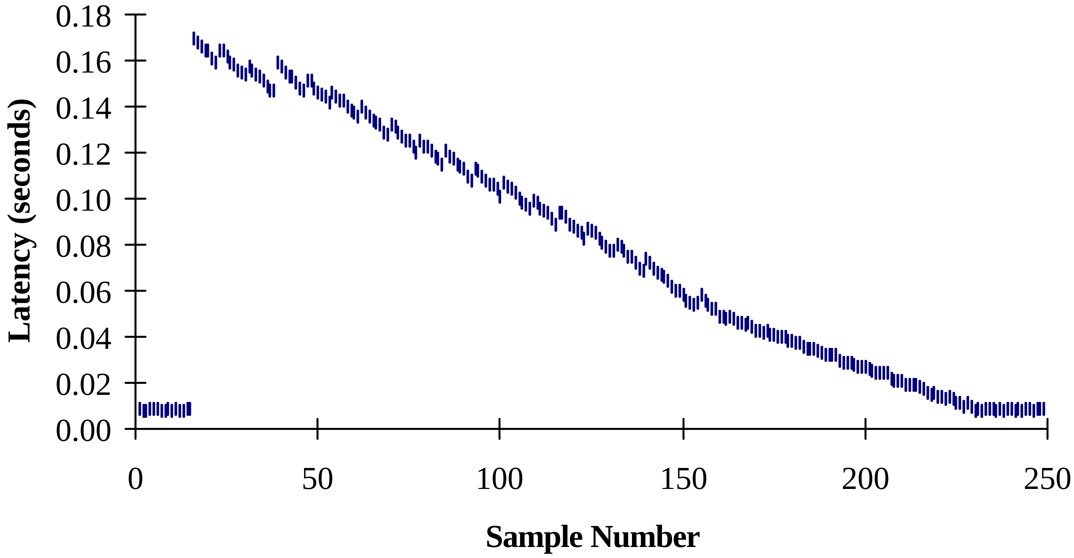


Figure 15. Largest Latency Event During the One-Way RTI Test

Figure 14 shows only the maximum latency observed for each combination of message rate and packet size. It does not indicate whether more than one latency event was observed, but closer examination of the data revealed multiple latency events in some cases. For example, for a message rate of 400 and a packet size of 101 in that figure, the event at sample #9677 that produced the maximum latency of 73 milliseconds was followed by a second event at sample #9715 with a maximum latency of 65 milliseconds. Figure 16 displays these latency events. Their close spacing within the 15000 messages transmitted for that rate and packet size probably is not a coincidence: it suggests that they may have had the same underlying cause.

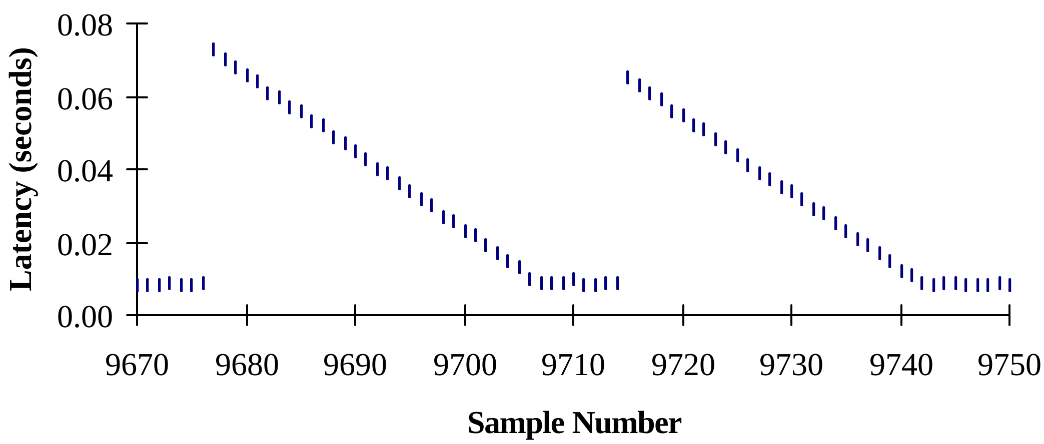


Figure 16. Closely-Spaced Latency Events During the One-Way RTI Test

The second latency event classification factor is the position of the sample number for the maximum latency relative to the first sample transmitted. For two out of the three latency events with a maximum latency greater than 70 milliseconds, the sample number at which the abrupt transition occurred was within the first 0.5% of the transmitted messages. This was also true for the smaller events in Figure 13 with maximum latencies of 57 and 37 milliseconds. The fact that the one-way RTI test showed both initial latency events and later, randomly occurring ones, combined with the similar features of the events, suggests that there may be separate causes for the events but a common mechanism for their time behavior. That mechanism may lie within the IRIX 6.3 TCP implementation.

7. Three-Node Test Description

These tests were designed to assist JADS in optimizing the performance of the RTI as well as the JADS EW Phase 2 test federation components. The major objective of these tests was to establish the performance baseline for the RTI and provide necessary feedback to JADS management as well as the RTI developers. Once the RTI version 1.3 performance baseline is determined by JADS testers, further testing, integration, and tuning of all federation components will be performed to support the Phase 2 implementation. These tests were the final benchmarks prior to the implementation and testing of actual Phase 2 test software federates with the AFEWES surrogate federate during August 1998.

The test environment expanded from the simple two-node configuration and used at least three and sometimes as many as six SGI O2 workstations (either 5000 or 10000 models) running IRIX 6.3 with GPS time code generators installed. The three-node test configuration in the EW test bed with six SGI computers is shown in Figure 17**.**

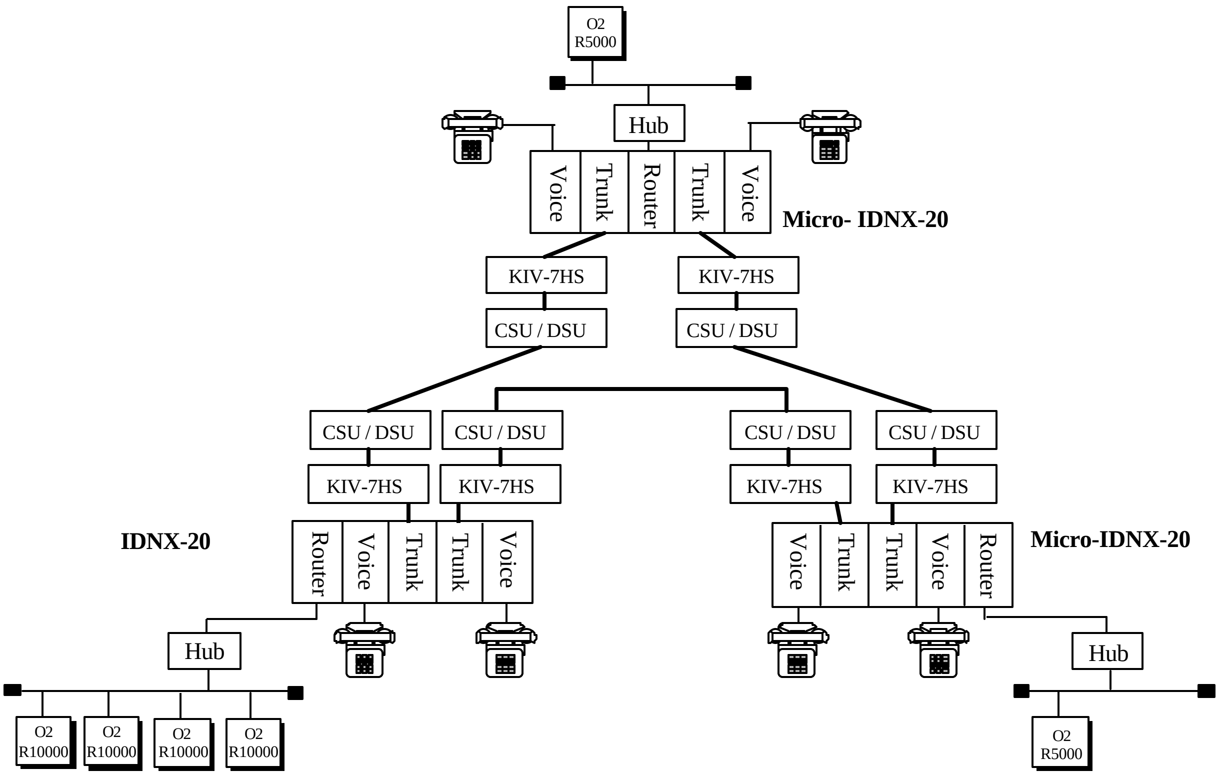


Figure 17. Three-Node RTI Test Configuration with Communications Devices

7.1 Multi-Federate Software for Three-Node Tests

After characterizing the network and the RTI in the simple, one-way tests, we wanted to determine whether the RTI would support the anticipated loads placed on it by the JADS federation. We wanted a test federate that could simulate these kinds of loads. The *testfed* federate was developed to satisfy these requirements. It can be executed on as many computers as necessary. The *testfed* federate accepts command line arguments that specify the characteristics of an instance of the federate. The user can specify these arguments:

1. Federate identification (ID) number (-f)
2. Duration of the test (-d)
3. Size of the attributes and interactions (-s)
4. Rate that attributes are published (-r)
5. Number of updates at the specified rate (-n)
6. Time the federate should wait before starting to publish at its specified rate (-w)
7. Whether interactions should be published (-i)
8. If the federate is the controller (-c)

During our tests, we ran *testfed* with the following options.

```
testfed -r20 -n11 -f1 -d300 -w5 -i        11 updates at 20 Hz with interactions
testfed -r20      -f2 -d300 -w5 -i        1 update at 20 Hz with interactions
testfed -r20 -n2  -f3 -d300 -w5 -i -c     2 updates at 20 Hz with interactions (controller)
```

There must be one and only one controller federate in the *testfed* federation. There is only one attribute and one interaction used by all federates. All federates subscribe to the attribute and the interaction.

7.2 Three-Node Three-Federate Tests with RTI 1.3-2EAV

For the three-federate test, we configured *testfed* on one computer to publish 11 attribute updates at 20 Hz (simulating the AFEWES federate). We configured another instance of *testfed* to publish 2 attribute updates at 20 Hz (simulating the federates at the JADS Albuquerque node). The third instance of *testfed* was configured to publish 1 attribute update at 20 Hz (simulating ACETEF). All three federates published interactions at approximately 1 Hz. The size of attributes and interactions was 121 bytes. Attributes were published best effort. Interactions were published reliable. We ran multiple tests with a duration of between two and five minutes.

Initially, we lost many attributes at the very beginning of a test. We surmised that there may be a problem with all federates beginning to publish at their specified rate all at the same time. Recent test results suggest that an initial burst of Ethernet collisions on an unswitched, half duplex 10BaseT LAN might have been responsible for this problem. We implemented the wait option (-w) to allow each federate to wait a certain amount of time before publishing at its regular rate. The wait option tells the federate to send attribute updates at 1 Hz for a specified number of seconds after the start time. Then, when the wait period expires, the federate publishes attribute updates at its normal rate. After we began using the wait option, the missing attributes at the beginning of the test were eliminated.

Some runs had only a few attributes lost with maximum latency less than 45 ms. Other runs had up to 100 attributes lost with maximum interaction latency of over 1 second. We ran three tests with all federates on the same unswitched, 10BaseT LAN. One of these tests had a maximum interaction latency of over 1.5 seconds.

7.3 Three-Node Six-Federate Tests with RTI 1.3-2EAV

After we leased three more SGI O2 computers, we ran a more realistic test with six federates on six computers on three network nodes separated by routers. The six-federate tests produced a wide variety of results. We had a few runs where only one or two best effort attributes were lost and the maximum latency was less than 50 ms. There were some runs that had up to 100 attributes lost and an occasional high interaction latency of between 1 and 8 seconds. There were also some runs that had federates that crashed. We reported these results to DMSO. Subsequently, DMSO found a software “bug” that limited the number of federates that could execute in a federation.

7.4 Three-Node Three-Federate Tests with RTI 1.3-2

RTI version 1.3-2 was the third version of release 1.3 we received and tested. We ran five tests with the same configuration: federate 1 publishes 11 attribute updates at 20 Hz with interactions sent at 1 Hz; federate 2 publishes 1 attribute update at 20 Hz with interactions sent at 1 Hz; and federate 3 publishes 2 attribute updates at 20 Hz with interactions sent at 1 Hz. All five tests had at least one federate with a maximum latency greater than 70 ms. The largest maximum latency value was 1.79 seconds. There were two tests that had a maximum over 250 ms.

7.5 Three-Node Six-Federate Tests with RTI 1.3-2

We ran two 5-minute tests and six 3-minute tests with six federates on three nodes. Since there were no runs that had federates that crashed, that problem appears to have been fixed by the RTI developer. However, in one of the 5-minute tests, all of the federates had an interaction maximum latency over 3 seconds (the worst was 10 seconds). Five of the six federates in the second test had interaction maximum latencies above 700 ms (the worst was 2.2 seconds).

7.6 Teleconferences

Because the RTI tests continued to produce runs with both attribute (best effort) and interaction (reliable) latencies above the JADS EW Test latency threshold of 70 milliseconds each way and some had interaction latencies exceeding 1 second, JADS began a series of weekly teleconferences with DMSO. These teleconferences provided a forum to discuss not only JADS test results, but the results of tests at Massachusetts Institute of Technology/Lincoln Laboratory (MIT/LL) and ACETEF where they are conducting tests with a similar network and JADS RTI test tools. This communication has produced some progress toward identifying possible causes of the latency problems and suggestions for how they might be resolved.

7.7 Recent Test Results

Testing during June, July, and early August produced these results.

- The initial and later latency events observed at JADS in “raw” TCP testing between two SGI O2s on an unswitched, half duplex, 10BaseT LAN have been reproduced at ACETEF using different SGI models and a high-speed, fiber distributed data interface (FDDI) LAN in addition to an ordinary Ethernet LAN, and at JADS after an upgrade to a switched, full duplex, 100BaseTX LAN. The exact cause of these events is not yet known, but their symptoms are thought to be due to start-up and/or transient response of the IRIX 6.3 TCP implementation.
- The symptoms of one type of 1-second-class interaction latency event have been traced to how the IRIX 6.3 TCP responds to the loss of two TCP packets over a short period of time (less than about 0.3 second), the first of which, in the four known cases, has been a 60-byte RTI heartbeat message. The root cause of this specific type of packet loss is not known, but JADS has provided test data and analysis procedures for these events to DMSO.

- The symptoms and cause of another type of 1-second or longer latency events have been traced to excessive Ethernet collisions on an unswitched, half duplex, 10BaseT LAN. It was noted and ACETEF confirmed that they used Ethernet switches to avoid such problems, JADS purchased and installed an 8-port Ethernet switch to upgrade the EW test bed to a switched, full duplex, 100BaseT LAN. Raw network and two-federate *testfed* tests with this new configuration have shown that the number of Ethernet collisions has been reduced to zero. Two-, three-, and seven-federate tests suggest that this upgrade may have eliminated or reduced the frequency of occurrence of the 1-second-class latency events significantly.
- ACETEF and DMSO reproduced, using the *testfed* tool, smaller latency events with maximum latency values in the 70 - 200 ms range.
- JADS three-federate tests with an RTI tick minimum value of 0.005 seconds (instead of the previous 0.0001 seconds) produced maximum latency values that were always less than 65 ms. Most of the time the maximum latency was less than 40 ms. Twenty 5-minute tests were run at expected JADS EW rates and sizes. All three federates were on the same LAN. These are the best results we've ever had in a series of three-federate tests. There were two tests that had high latency (in the hundreds of ms). This was because someone logged onto one of the test machines during the run.

7.8 Background Research on TCP and TCP Implementations

In another effort to identify and resolve these latency problems, JADS has studied the TCP literature for pertinent information. This research provided the clues that explained the symptoms of the 1-second-class latency events caused by lost TCP packets. It has also revealed considerable differences between vendors in their implementations of the TCP protocol as described in its two main RFCs (References 3 and 4).

8. Lessons Learned

8.1 Time-to-Live

In the initial tests we performed with RTI 1.0-2, best effort traffic was not received at any computer on a different LAN. Using the network packet "sniffer" tool to look at the network data packets, one of the JADS network engineers discovered that the time-to-live (TTL) value was set to 1. A packet's TTL indicates how many hops it can take before it is discarded by the network. A value of 1 does not allow a packet to exit the LAN, i.e., to pass through a router to reach a system on another LAN or a WAN. Hence, a federation running with RTI 1.0-2 out of the box would not allow federates to communicate best effort traffic outside of a LAN. Using the JADS 2-node network configuration (shown in Figure 4) required network data packets to cross from one LAN through the routers (Micro-IDNX-20) to reach the test federate on another LAN mirroring the JADS EW Phase 2 network architecture. JADS was provided a new library from DMSO that allowed us to use RTI 1.0-2 across our network communications gear. Subsequent versions of the RTI provide for a user-defined parameter value in the RTI initialization data (RID) file to set the TTL.

8.2 TCP No Delay and the Nagle Algorithm

Prior to RTI version 1.3-2, the RTI ran with default setting for the TCP_NODELAY socket option. On the SGIs, the default value for this option is FALSE. This means that the Nagle algorithm will be in effect for both attribute and interaction data sent reliable. If data is published using reliable transportation at data rates at or above 5 Hz, then the latency of the data is increased as illustrated in Figure 6 showing the initial TCP test matrix results. As a result of sharing this information with RTI developers, RTI version 1.3-2 sets the TCP_NODELAY option to TRUE, disabling the Nagle algorithm.

8.3 Tick

As JADS implemented and experimented with the RTI tick function during initial test runs with each RTI release, we learned how important it is to understand how tick works in its various forms in order to tune a federation properly. Each federation and its architecture is different, and it will require some experimentation by the federation developers to find the optimum use of tick. The tick function is how a federate transfers process control to the RTI so it can do its work. Each federate must constantly tick the RTI or nothing will happen in the federation. There are two variations to tick: one has no arguments (tick [ ]), while the other has a minimum and a maximum argument (tick [min, max]) with units of seconds for both. When a federate calls the tick function with no arguments, tick empties its queue before it returns to the federate. This could starve the federate from getting its necessary processor time.

If a federate calls tick with values for the minimum and maximum arguments, it will stay at least the amount of time specified by the minimum argument, but no longer than the maximum argument. If the RTI empties its queue before the minimum time elapses, it will try to "sleep" for the rest of the time. On an SGI, this is a problem because the minimum sleep time is 10 ms (the functions sginap and select behave similarly). Thus, if the federate specifies a minimum value of 10 ms and the RTI uses 9 ms to do its work, on an SGI it will "sleep" for an additional 10 ms.

On the other hand, if the federate specifies zero or some small number for the minimum, the RTI will not "sleep." But this can cause the federate/RTI to use as much as 90% of the central processing unit (CPU). We benefited greatly from open communication with DMSO about features of tick and verifying the results we obtained from different settings. Unfortunately, we did not find any source of documentation for tick features and tuning ideas. We advised DMSO that this information would be very beneficial to all but the casual RTI user.

8.4 Initial Publication Rates

When a federate starts, we found that it is best if it publishes some initial data at low data rates to set up the network. In the JADS tests, with three federates (one that published 11 updates at 20 Hz, one that published 2 updates at 20 Hz, and a third that published one update at 20 Hz), best effort data was lost and reliable data had high latencies in the initial burst of data. When we added a 5-second delay at the start during which the federates published data at 1 Hz, these start-up problems were eliminated. Excessive Ethernet collisions may have caused the lost best effort data, while the start-up and transient behavior of the IRIX 6.3 TCP implementation may have caused or contributed to the reliable data high latencies.

8.5 Fast Malloc

SGI provides an IRIX library that includes a faster version of the malloc function, which is used to dynamically allocate memory. To use this library, it must be linked with federate software with the lmalloc option. In an attempt to make it as efficient as possible, the JADS RTI logger was linked with this library. While running RTI tests linked with the logger, the federate would crash after it resigned from the federation. After speaking with DMSO, they said they were aware of problems using this library and recommended not using it.

8.6 Optimize Factors You Can Control

Distributed simulations are, by their very nature, complicated, and those conducting them may not have control over all factors that may affect simulation performance. Sometimes, though, there are factors that can not only be controlled, but optimized, and at low cost. The upgrade of the JADS EW test bed from an unswitched, half duplex, 10BaseT LAN to a switched, full duplex, 100BaseTX LAN cost only about $500, and the equipment was identified, purchased, received, installed, and in use within one week. Test results demonstrated that this simple device significantly improved test bed performance, and it may have eliminated or reduced in frequency some of the large latency problems.

8.7 Don't Assume All Vendor TCP Implementations Are the Same

Since HLA-compliant federations using the current RTI must communicate via the internet user datagram protocol (UDP), TCP, and IP protocols, their performance is constrained by **both** the protocols themselves and by specific vendor implementations of those protocols. Naively, a federation developer might assume that, since these protocols have been in existence for many years and are currently used by literally tens of millions of computers worldwide, most vendor implementations would be almost identical and would conform closely to the same sets of specifications. Unfortunately, as the analysis team's research of the TCP literature has shown, that is definitely not true (see References 5, 6, and 7).

In particular, SGI's IRIX 6.3 TCP which is probably based on the Berkeley Software Distribution (BSD) Network Releases (such TCPs are sometimes called "BSD-derived implementations"), may differ significantly from the Solaris 2.5 and 2.5.1 TCPs developed by Sun Microsystems. Since the current RTI is being developed, tested, and maintained primarily on systems using Solaris and running over a single LAN, but JADS, ACETEF, and AFEWES use IRIX-based systems on several LANS that must be connected by three WANs, it no longer seems surprising that problems occurred during RTI testing. JADS probably should be prepared to encounter further network-related RTI problems in the near future. Use of dissimilar platforms will be an even greater challenge to future HLA users.

9. Summary

This report documents the JADS tests of the HLA RTI conducted between March and early August 1998. During this time frame, the following versions of the RTI were tested:

RTI Version	Date Released
1.0-2	February 1998
1.3b	3 April 1998
1.3-2 EAV	15 May 1998
1.3-2	15 June 1998

Based upon the latency values measured in early August for the most recent RTI software release, further tests may need to be conducted when resolution of the remaining latency problems is accomplished by DMSO. As documented, much has been accomplished and learned by both JADS and DMSO's RTI team, based upon this effort. The progress made and lessons learned thus far represent a significant advance, but the results do not yet satisfy JADS criteria for success. DMSO continues to provide significant support to address RTI problems as they are discovered.

DMSO released the "final" version of RTI version 1.3-2 for IRIX 6.3 SGI workstations in July 1998. JADS will assess with DMSO when further versions of the RTI software will be tested.

10. References

1. Mills, David L. *Network Time Protocol (Version 3) - Specification, Implementation and Analysis*, Network Working Group, Request for Comments 1305, University of Delaware, March 1992.

2. Stevens, W. Richard. *TCP/IP Illustrated, Volume 1 - The Protocols*, Section 19.4, "Nagle Algorithm," Addison Wesley, 1994, pp. 267-273.

3. *Transmission Control Protocol - DARPA Internet Program Protocol Specification*, Information Sciences Institute, Request for Comments 793, University of Southern California, September 1981.

4. Braden, R., Editor. *Requirements for Internet Hosts – Communication Layers*, Network Working Group, Request for Comments 1122, Internet Engineering Task Force, October 1989.

5. Fall, Kevin and Sally Floyd. *Comparison of Tahoe, Reno, and Sack TCP*, Network Research Group, Lawrence Berkeley Laboratory, December 2, 1995.

6. Paxton, Vern. *Automated Packet Trace Analysis of TCP Implementations*, Network Research Group, Lawrence Berkeley Laboratory, June 23, 1997.

7. Dawson, Scott, Farnam Jahanian and Todd Milton. *Experiments on Six Commercial TCP Implementations Using a Software Fault Injection Tool*, Real-Time Computing Laboratory, Department of Electrical Engineering and Computer Science, University of Michigan, 1997.

Attachment 1 Acronyms and Abbreviations

A/C	aircraft
ACETEF	Air Combat Environment Test and Evaluation Facility, Patuxent River, Maryland; Navy facility
ACK	acknowledgment packet
ADRS	Automated Data Reduction Software
ADS	advanced distributed simulation
AFEWES	Air Force Electronic Warfare Evaluation Simulator, Fort Worth, Texas; Air Force managed with Lockheed Martin Corporation
ALQ-131	a mature self-protection jammer system; an electronic countermeasures system with reprogrammable processor developed by Georgia Technical Research Institute
AMG	Architecture Management Group
API	application program interface
ATEWES	Advanced Tactical Electronic Warfare Environment Simulator
BSD	Berkeley Software Distribution
CPU	central processing unit
CSU	channel service unit
DMSO	Defense Modeling and Simulation Organization, Alexandria, Virginia
DoD	Department of Defense
DSM	digital system model
DSU	data service unit
EAV	early access version
env	environment
EW	electronic warfare
FDDI	fiber distributed data interface
FOM	federation object model
GPS	global positioning system
HITL	hardware-in-the-loop (electronic warfare references)
HLA	high level architecture
Hz	hertz
I/F	interface
I/O	input/output
IADS	Integrated Air Defense System
ID	identification
IP	internet protocol
IRIG	Inter-Range Instrumentation Group
IRIX	operating system for the Silicon Graphics, Inc.
JADS	joint advanced distributed simulation or Joint Advanced Distributed Simulation, Albuquerque, New Mexico
JETS	JammEr Techniques Simulator

km	kilometer
LAN	local area network
LL	Lincoln Laboratory
MHz	megahertz
MIT	Massachusetts Institute of Technology
ms	millisecond
NTP	network time protocol
OAR	open air range
PC	personal computer
RF	radio frequency
RFC	request for comment
RID	RTI initialization data
RTC	reference test condition
RTI	runtime infrastructure
SGI	Silicon Graphics, Inc.
SISO	Simulation Interoperability Standards Organization
SPJ	self-protection jammer
STIM	radio frequency stimulator
SUT	system under test
T&E	test and evaluation
T-1	digital carrier used to transmit a formatted digital signal at 1.544 megabits per second
TAMS	Tactical Air Mission Simulator
TCF	test control federate
TCP	transmission control protocol
TTH	terminal threat hand-off federate
TTL	time-to-live
UDP	user datagram protocol
USDA&T	Under Secretary of Defense for Acquisition and Technology
VV&A	verification, validation, and accreditation
WAN	wide area network

Attachment 2 JADS EW Federation Execution Planners' Workbook

Host Table

	Hardware	Operating System	Memory available to RTI (MB)	Total CPU Available to Federation and RTI Combined (% CPU Cycles)	% CPU Available to RTI	Notes (Use to explain how % CPU available to RTI derived)
1 - PLATFORM	SGI 02 R5000	6.3	25	90%	25%	This federate will host the RTI Ambassador, Federate and Logger. These will be executed as one process. Since an idle machine with no processes has about 98% CPU free we are allotting 10% CPU usage for time synchronization and other functions.
2 - RF ENV	SGI 02 R10000	6.3	25	90%	25%	See Federate 1. In addition, this federate will host the RTI Exec and the FEDEX.
3 - DSM-JAMMER	SGI 02 R5000	6.3	25	90%	25%	See Federate 1.
4 - TST CTRL FAC	SGI 02 R10000	6.3	25	90%	25%	See Federate 1.
5 - THREATS	SGI Challenge	6.2	96	90%	25%	6 CPUs available. Only one will be used by the Federate and the RTI. The processing time for the federate should be greater than the RTI. Therefore, 25% selected for RTI use with the other 65% for the federate. The other CPUs will be for other apps
6 - TERM THRT HANDOFF	SGI 02 R5000	6.3	25	90%	25%	See federate 1

NOTE:
Complete one of these tables for each Federation execution